SOFTVERSKI SISTEM ZA AUTOMATSKO ODREĐIVANJE SEMANTIČKE SLIČNOSTI KRATKOG TEKSTA

Davor Jovanović, Bojan Furlan, Boško Nikolić, Elektrotehnički fakultet u Beogradu, Univerzitet u Beogradu

mdmdavor@yahoo.com , bojan.furlan@etf.bg.ac.rs , bosko.nikolic@etf.bg.ac..rs

Sadržaj – *U radu je opisan softeverski sistem za semantičku analizu kratkog teksta. Pored toga objašnjeni su osnovni principi u razvoju sistema uključujući algoritme za leksičko i semantičko poklapanje, ali i algoritam koji objedinjuje ova dva algoritma u cilju dobijanja najboljih rezultata. Izložen je pregled u fazi razvoja aplikacije, kao i sama evaluacija dobijenih rezultata aplikacije. Takođe, na kraju se pored dobijenih rezultata poseban akcenat stavlja na dalji rad i moguću primenu postojećih metoda za srpski jezik.*

1. Uvod

Danas se veliki deo elektronski dostupnih informacija bazira na Webu i sličnim medijima, i predstavljen je kratkim tekstualnim navodima. Prilikom automatske obrade ovakve vrste teksta, uloga semantičke sličnosti između njegovih delova je sve veća.

Zato se u ovom radu razmatra obrada kratkog teksta. Primenjuje se izračunavanje sličnosti između dve rečenice ili dva pasusa pomoću postojećih algoritama za procesiranje tekst korpusa, pravljenjem semantičkog prostora reči i njegove primene u okviru srpskog jezika.

U otvorenoj literaturi se izdvajaju algoritmi za leksičko poklapanje i algoritmi za semantičko poklapanje među rečima, čija je uloga kreiranje semantičkog prostora. U narednim odeljcima se razmatraju mogućnosti realizacije i primene ovih algoritama.

Zatim je opisan SSPACE paket koji je deo *Google airhead* projekta, a koji se kao takav koristi u izboru i primeni algoritma za kreiranje semantičkog prostora. Nakon toga sledi prezentacija algoritma koji podjednako uzima u obzir i semantičko i leksičko poklapanje i koji kao konačan rezultat vraća meru semantičke sličnosti između dve rečenice.

Eksperimentalnim putem je pokazano da opisani pristup merenja semantičke sličnosti dovodi do poboljšanja dosadašnjih dostupnih rezultata. Dobijeni rezultati se odnose na proveru semantičke sličnosti između parova kratkih engleskih rečenica, ali se poseban akcenat stavlja na primenu istih metoda za srpski jezik. Nakon izvršene evaluacije dobijenih rezultata dat je zaključak rada sa razmatranjem moguće nadogradnje sistema.

2. Opis problema

Analiza kratkog teksta predstavlja temu velikog broja istraživanja, kako ranijih godina, tako i današnjih projekata. Prvi pristup rešavanju ovog problema je bio računanje leksičke sličnosti između reči, dok se noviji algoritmi zasnivaju na većoj upotrebi semantičke sličnosti. Postoji mogućnost i zajedničke upotrebe ove dve vrste algoritama, kako bi se dobili najbolji rezultati.

Za određivanje leksičke sličnosti među rečima moguće je koristiti tri modifikovane verzije *LCS* (*Longest Common Subsequence*) algoritma uz odgovarajuće normalizacije [1]: $MCLCS_1$ (*maximal consecutive longest common subsequence starting at character 1*) i $MCLCS_N$ (*maximal consecutive longest common subsequence starting at any character N*). $MCLCS_1$ traži najdužu moguću sekvencu poklapanja kraćeg i dužeg stringa, ali počev od pozicije prvog karaktera, dok $MCLCS_N$ traži najdužu moguću sekvencu poklapanja kraćeg i dužeg stringa, ali počev od bilo koje pozicije karaktera.

Za određivanje semantičke sličnost koriste se metodi pomoću baze sinonima-rečnika, gde se mera sličnosti određuje operacijom prolaska kroz graf povezanosti. Druga vrsta metoda se oslanja na algoritme koji meru semantičke sličnosti među rečima uče iz velike kolekcije teksta (*corpus-based measures*), koja je i primenjena u ovom radu.

Korpusi predstavljaju velike kolekcije teksta. Mogu se isporučivati u vidu jednog velikog fajla, ili u više odvojenih, i uglavnom su u *.xml* ili *.txt* formatu. Postoje i besplatni korpusi engleskog jezika, kao što je *American National Corpus* sa 15 miliona reči, *USENET* corpus sa oko 20 milijardi uglavnom engleskih reč i *Wikipedia* corpus, koji je višejezičan.

Algoritmi zasnovani na učenju na bazi korpusa pokušavaju da od prečišćenog korpusa stvore semantički prostor na osnovu distribucije reči unutar samog korpusa. U tom prostoru svaka reč ima svoj konteksni vektor, a relacija među tim vektorima zapravo predstavlja semantičku sličnost među rečima. Ovaj zaključak motivisan je distribucionalnom hipotezom koja tvrdi da reči sa sličnijim značenjem imaju tendenciju da se pojavljuju u sličnim kontekstima [2].

U literaturi se mogu naći mnogobrojni algoritmi koji se baziraju na ovom konceptu, sa varijacijama u normalizaciji elemenata matrice, tehnici postprocesiranja, trenutku konstrukcije matrice, kao i veličini konteksta i tipu konteksta u kome se posmatra frekfencija pojavljivanja reči. Najpoznatiji predstavnici su LSA (*Latent Semantic Analysi*,) [3], HAL (*Hyperspace Analogue to Language*) [4], COALS (*Correlated Ocurrence Analogue to Lexical Semantic*) [5] i RI (*Random Indexing*) [2]. Nabrojani algoritmi koriste *co-ocurrencce* matrice, na osnovu kojih se mogu izvesti zaključci o sličnosti između pojedinih vektora i na taj način odrediti i sličnost između reči.

Samo formiranja vektorskog prostora iziskuje dosta resursa, stoga se sprovode tehnike predprocesiranja u cilju smanjenja ukupnog broja reči, pa i postprocesiranje u cilju redukcije dimenzije vektora. Najpoznatija tehnika za redukciju dimenzija vektora je SVD (*Singular Value Decomposition*). SVD je algebarska operacija koja nad zadatom matricom vrši faktorizaciju i dekompoziciju, što omogućuje različite aproksimacije i odbacivanje nekih kolona, a samim tim i redukciju dimenzije vektora.

Nakon analize, odlučeno je da se testiranje sprovede za algoritam COALS, koji je razvijen na MIT univerzitetu, kao poboljšanje HAL algoritma. Sam algoritam vrši implementaciju *co-ocurrencce* matrice tako što definiše n-reči u okviru prozora (*n-words-window*), gde n predstavlja broj reči u okviru jednog prozora. Zatim se posmatra proizvoljna reč **Wr** iz vrste i proizvoljna reč **Wc** iz kolone, pa se se za ceo korpus sumira udaljenost reči **Wc** od reči **Wr**. S obzirom da je preporuka da se za kontekst pojavljivanja koristi *4-word-window*, posmatraju se po četri reči i ukoliko je **Wc** reč odmah do **Wr** reči dobijamo vrednost 4, ukoliko je udaljena dve reči dobija se vrednost 3 itd. Nakon formiranja inicijalne matrice primenjuje se normalizacija svih vrednosti elemenata matrice. Nakon toga

se dobija matrica čije vrednosti imaju opseg u granicama od -1 do 1. Sve negativne vrednosti se postavljaju na nulu, dok se nad pozitivnima sprovodi operacija dobijanja korena. U konačnoj matrici vrste predstavljaju konačne vektore. Opciono se u okviru COALS algoritma može uključiti i SVD postprocesiranje u cilju redukcije dimenzija vektora.

SSPACE paket [6] je deo *Google airhead* projekta i predstavlja *open-source* paket koji je implementiran u Java programskom jeziku. Ovaj paket poseduje brojne implementirane algoritme za određivanje semantičke sličnosti na bazi korpusa. Pored toga SSPACE paket nudi veoma korisne klase koje omogućuju kreiranje sopstvenog algoritma, kao i odgovarajuće testove. Među brojnim implementiranim algoritmima se nalazi i COALS koji je iskorišćen za implementaciju opisanog sistema. Jedinstven interfejs omogućuje integraciju sa proizvoljnom aplikacijom koja ima potrebe za semantičkom analizom. Pomoću SSPACE paketa i izabranog algoritma, se generiše odgovarajući semantički prostor.

3. Algoritam određivanja sličnosti između dve rečenice

Algoritam određivanja semantičke sličnosti između dve rečenice se podjednako oslanja i na leksičko i na semantičko poklapanje [1]. Sam rad algoritma se sastoji iz 6 koraka:

I. Vrši se eliminacija svih interpukcijskih znakova i uklanjaju se sve tzv. stop-reči. Zatim se rečenice razbijaju na tokene i tako se dobija:

$$P = \{p_1, p_2, \dots, p_m\} \quad R = \{r_1, r_2, \dots, r_n\}$$

Gde P predstavlja niz koji sadrži *m* tokena (reči), a R *n* tokena (reči), pri čemu važi da je *m*<=*n*.

II. Formira se broj δ, koji predstavlja broj potpunih leksičkih poklapanja skeniranjem nizova P i R za svako *pi* = *rj*, gde važi *i*<=*m; j*<=*n*. Taj broj, δ predstavlja broj tokena u P koji se potpuno poklapaju sa rečima u R. Mora da važi: δ<=m.
Nakon toga uklanjaju se svi tokeni koji se poklapaju iz nizova P i R i dobijaju dva nova niza:

$$P = \{p_1, p_2, \dots, p_{m-\delta}\} \quad R = \{r_1, r_2, \dots, r_{n-\delta}\}$$

III. Konstruiše se matrica tipa (m-δ)x(n-δ). Način konstrukcije je da se preračunava za svaki token iz oba niza P i R vrednost *leksičkog poklapanja* α_i to za svaku reč iz niza P i R i na taj način se formira matrica.

$$M_1 = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1j} & \dots & \alpha_{1(n-\delta)} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2j} & \dots & \alpha_{2(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \dots & \alpha_{ij} & \dots & \alpha_{i(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{(m-\delta)1} & \alpha_{(m-\delta)2} & \dots & \alpha_{(m-\delta)j} & \dots & \alpha_{(m-\delta)(n-\delta)} \end{pmatrix}$$

Slika 1 Formiranje matrice *leksičkog poklapanja*

IV. Formira se i druga matrica, gde njeni elementi predstavljaju meru semantičke sličnosti. Za određivanje semantičke sličnosti mogu se koristiti različiti alati, kao što su na primer alati iz SSPACE paketa. Tako se konstruiše matrica *semantičkog poklapanja* čije su dimenzije (m- δ)x(n- δ). Za svaki token iz oba niza P I R proračunava se semantička sličnost. Neka je to vrednost β_{ij}. Sada se vrednost β_{ij} postavlja u vrstu *i* i kolonu *j* matrice.

$$M_2 = \begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1j} & \dots & \beta_{1(n-\delta)} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2j} & \dots & \beta_{2(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta_{i1} & \beta_{i2} & \dots & \beta_{ij} & \dots & \beta_{i(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta_{(m-\delta)1} & \beta_{(m-\delta)2} & \dots & \beta_{(m-\delta)j} & \dots & \beta_{(m-\delta)(n-\delta)} \end{pmatrix}$$

Slika 2 Formiranje matrice *semantičkog poklapanja*

V. U sledećem koraku se konstruiše udružena matrica (*joint matrix*) koja je istih dimenzija kao i M1 i M2. Udružena matrica se formira po sledećoj jednačini:

$$M \leftarrow \psi M_1 + \varphi M_2$$

gde ψ i φ predstavljaju odgovarajuće težine za leksičko poklapanje i semantičko poklapanje, respektivno i važi $\psi + \varphi = 1$.

$$M = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \dots & \gamma_{1j} & \dots & \gamma_{1(n-\delta)} \\ \gamma_{21} & \gamma_{22} & \dots & \gamma_{2j} & \dots & \gamma_{2(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_{i1} & \gamma_{i2} & \dots & \gamma_{ij} & \dots & \gamma_{i(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_{(m-\delta)1} & \gamma_{(m-\delta)2} & \dots & \gamma_{(m-\delta)j} & \dots & \gamma_{(m-\delta)(n-\delta)} \end{pmatrix}$$

Slika 3 Udružena matrica

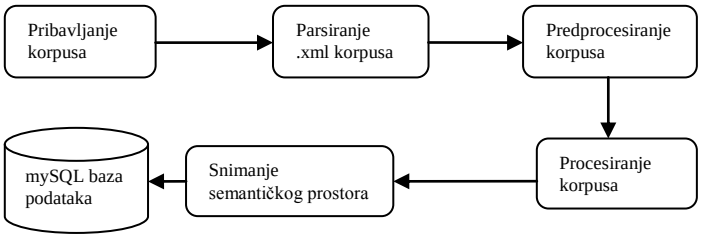
Nakon konstrukcije udržene matrice, nalazi se maksimalan element γ_{ij} i uklanjaju se iz matrice sve elemente iz vrste *i*, i kolone *j*. Pri tom, mora da važi γ_{ij}>=0. Pronađeni maksimalni element γ_{ij} dodaje se u listu, na primer ρ. Proces se ponavlja sve dok u matrici ima elemenata ili dok svi preostali elementi nemaju vrednost 0. Na kraju se dobija lista ρ sa maksimalnim elementima po vrstama i kolonama.

VI. Za dobijanje konačne sličnosti S(P,R), između rečenica P i R, koristi se sledeća jednačina:

$$S(P,R) = \frac{\left(\delta + \sum_{i=1}^{|\rho|} \rho_i\right) * (m+n)}{2mn}$$

4. Realizacija softverskog sistema

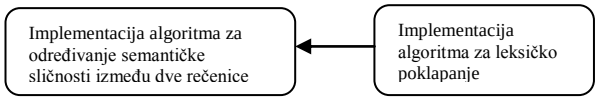
Slika 4 ilustruje način realizacije softverskog sistema koji je iskorišćen za evaluaciju i poređenje navedenih algoritama.



Slika 4 Faze u razvoju aplikacije

Prva faza obuhvata pribavljanje samog korpusa. Za potrebe istraživanja veličina RAM memorije izabranog računarskog sistema je bila 4GB, i izabrani korpus imao maksimalnu

veličinu od 1.4GB. Korpus je bio *wiki-abstract.xml* i sadržao je samo abstrakte svih članaka u okviru Wikipedia servisa.
Kako je primenjeni korpus bio u obliku jednog fajla i u .xml formatu, bilo je potrebno prilagoditi odgovarajući parser koji će ukloniti *xhtml* komande i na taj način dobiti tekst od interesa. Naredni korak je predprocesiranje korpusa, gde je izvršena tehnika stemming, zatim uklanjanja reči koje sadrže brojeve i karaktere koji nisu sastavni deo engleskog alfabeta, kao i samih brojeva i engleskih stop reči.
Stemming operacija predstavlja dobijanje korena reči. Na primer u engleskom jeziku: fisher, fished, fishing imaju istu korenu reč fish. Ovde se može uočiti da se stemming operacijom smanjuje veličine matrice, pa samim tim i štedi memorija kao ključni resurs u narednom koraku. Nakon predprocesiranja korpus je spreman za procesiranje u smislu kreiranja semantičkog prostora, pomoću nekog od postojećih algoritama iz SSPACE paketa. Za datu implementaciju izabran je COALS algoritam, a za postprocesiranje u cilju redukcije dimenzije vektora izabrana je SVD algebarska operacija.
Na kraju, snimanje semantičkog prostora predstavlja operaciju upisivanja dobijenih rezultata na spoljašnjoj memoriji u jednom od SSPACE formata. Primenjeno je snimanje za sparse mode i u tekstualnom formatu. Dobijeni tekstualni fajl ja sadržan od parova jedinstvenih reči i njima pridruženih vektora. S obzirom da standardne metode otvaranja fajla i čitanje linije po linije utiču na degradiranje performansi, odlučeno je da se kreira baza podataka koja sadrži samo jednu tabelu sa poljima jedinstvene reči i njenog vektora. Nakon konstrukcije baze izvršeno je odgovarajuće indeksiranje u cilju ubrzanja pretrage nad rečima.



Slika 5 Konačna faza u razvoja aplikacije

Zatim je potrebno implementirati ranije objašnjene algoritme. Naime, u bazi postoje reči i njihovi odgovarajući vektori. Pri konstrukciji semantičke matrice koriste se dve operacije: čitanje vektora iz baze za odgovarajuće dve reči i njihovo kosinusno upoređivanje.

5. Dobijeni rezultati

Za potrebe evaluacije i testiranja korisšćena je kolekcija parova rečinica MSRPC (*Microsoft Shared source Parphase Corpus*, [7]). MSRPC korpus se sastoji od 5801 para rečenica. Svaki par rečenica je ocenjen (data je ocena semantičke sličnosti) od strane dvojice sudija. Njihova ocena je bila binarna, gde ocena jedan ukazuje da su rečenice slične, a nula obrnuto. Slučajeve kada dolazi do neslaganja ocena sudija, rešava treći. Posle konačne ocene 3900 (67%) rečenica, od ukupno 5801, je bilo semantički ekvivalentno. Sličnost u ocenama između troje sudija iznosi 83%, pa se može zaključiti da je pouzdanost test primera zadovoljavajuća.
Radi evaluacije MSRPC korpus je podeljen na dva fajla: *train* i *test*. *train* fajl sadrži veći skup rečenica i služi za određivanje optimalne vrednosti praga (*threshold score*). Prag vrednosti se kreće u granicama od 0 do 1 i određuje granicu ocene semantičke sličnosti, tako da svi rezultati koji budu iznad te vrednosti govore da su rečenice semantički slične, dok vrednosti ispod praga govore da rečenice nisu ekvivalentne. Kada se odredi prag vrednosti, za tu vrednost se sprovodi analiza nad *test* fajlom i ukoliko se dobiju očekivani rezultati, prag vrednosti je usvojen čime je analiza završena.

Prag vrednosti	Preciznost
0.4	67.75%
0.5	69.27%
0.589	71.33%
0.6	71%
0.7	67.72%
0.8	57.4%
0.9	40.37%

Tabela 1. Određivanje optimalne vrednosti praga u *train* skupu podataka

Prag vrednosti	Preciznost
0.4	66.7%
0.5	69.4%
0.589	70.32%
0.6	70.1%
0.7	67.8%
0.8	58%
0.9	41.4%

Tabela 2. Evaluacija optimalne vrednosti praga nad *test* skupom podataka

Prilikom odluke da li su dve rečenice semantički slične ili ne na osnovu Tabela 1 i 2 može se zaključiti da je idealni prag 0.589 i da sa tim pragom možemo biti 70.32% sigurni u tačnost odluke sistema. Sa druge strane treba uzeti u obzir nesavršenost podataka nad kojima je vršena evaluacija, jer je rečeno da je korelacija u rezultatima sudija bila 83%.

6. Primena rezultata evaluacija na srpski jezik

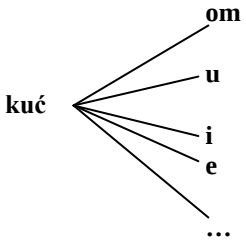
Na osnovu priloženih rezultata može se zaključiti da, iako su algoritmi primenjeni na manjem korpusu reči, dobijeni rezultati su zadovoljavajući. Shodno tome zaključak je da se za veće korpuse istom tehnikom mogu dobiti značajno bolje performanse.
Autori su razmatrali i pitanje da li se priložena tehnika može iskoristiti za srpski jezik. Ako se analizira realizacija sistema, može se primetiti da bi jedina promena bila u pribavljanju korpusa i procesu predprocesiranja. U slučaju pribavljanja korpusa ne postoji problem, s obzirom da *Wikipedia* isporučuje isti format korpusa i za srpski jezik. Pri tom, količina podataka je znatno manja, ali se može iskoristiti na zadovoljavajući način. Kod procesa predprocesiranju najveći problem predstavlja primena stemming tehnike, tj. postupak dobijanja korena neke reči. Neka u semantičkom prostoru za srpski jezik postoje odvojene reči: kući, kuća, kućom, kućo. Sve ove reči imaju isti koren: kuća. Zato se može zaključiti sve imenice treba svesti na padež u nominativu. Na primer: „Upravo sam krenuo kuća“ jeste gramatički neispravan, ali ne utiče na performanse semantičke analize, već ih samo poboljšava tako što algoritam posmatra jedinstvenu reč kuća u različitim kontekstima, a ne odvojeno. Trenutno za problem primene stemming tehnike za srpski jezik postoji jako mali broj predloženih rešenja.
Generalno u realizaciji stemming tehnike za srpski jezik mogu se primeniti dva pristupa:

- Pristup na bazi rečnika
- Pristup na bazi čistog algoritma

Drugi pristup je zbog kompleksnosti srpske gramatike teško izvodljiv, ali to ne isključuje mogućnost implementacije.

Neka su definisana dva pojma: stemm pravila i baza sufiksa. Tada je moguće:

- Pribavljanje ogromnog (u smislu fonda reči) rečnika srpskog jezika u elektronskom obliku
- Pribavljanje korpusa srpskog jezika (moguće rešenje je korpus koji dostavlja *Wikipedia*)
- Izvršiti skeniranje korpusa (podrazumeva se analiza svake reči koja se pojavljuje u korpusu) i za svaku reč traži se najsličnija (leksičko poklapanje-MCLCS1) u rečniku.
- Uzima se reč iz rečnika sa najboljim leksičkim poklapanjem i skenirana reč iz korpusa. Pronalazi se sufiks (reč u rečniku: **reka**, reč iz korpusa: **rekom**, u ovom slučaju sufiks je **om**) i ubacuje u bazu sufiksa.
- Za svaki sufiks neke reči vrši se prebrojavanje, tj. frekvencije njegovog pojavljivanja.
- Treba imati u vidu da se popunjavanje baze sufikasa vrši pri dobijanju svake reči iz korpusa i upoređivanjem sa rečima u rečniku srpskog jezika.
- Kada se popuni baza sufikasa, vrši se redukcija onih sufiksa koji imaju malu frekvenciju pojavljivanja. Nakon toga se za svaku reč kreira stablo sledećeg oblika:



Nakon priložene metode kreirala bi se baza stabala, koja zapravo predstavlja stemming pravila, gde se svako stablo sastoji od korene reči i odgovarajućeg sufiksa. Stemming algoritam se sada svodi na pretraživanje baze za svaku reč i kretanje po stablu do potpunog poklapanja. Ukoliko dođe do poklapanja, odgovarajući sufiks se otklanja i time se završava stemming operacija.
Na kraju bi trebalo sprovesti kreiranje ili pak prevođenje MSRPC-a na srpski jezik u cilju evaluacije.

7. Zaključak

U ovom radu opisan je razvoj i evaluacija jednog softverskog sistema za određivanje semantičke sličnosti kratkog teksta. Određivanje sličnosti vrši se poređenjem svih parova reči. Prezentovani pristup pored računanja semantičke sličnosti dve reči, zasnovanog na COALS algoritmu, koristi i meru njihove leksičke sličnosti, čime se dobijaju precizniji rezultati. Takođe, razmatrano je i pitanje da li se priložena tehnika može iskoristiti za srpski jezik.
S obzirom da veliki deo informacija dostupnih danas, na Vebu i drugde, se sastoji od kratkih tekstualnih isečaka (npr. sažetaka naučnih dokumenata , natpisa slika ili opisa proizvoda) upotreba ovakvog sistema je višestruka. S druge strane, jedna od mogućnost primene datog sistema je i u edukaciji pri automatskom ocenjivanju kratkih odgovora studenata [8].
Na kraju, sprovedena evaluacija pokazuje da dobijeni rezultati odgovaraju rezultatima prezentovanim u [1], s tim što za određivanje semantičke sličnosti dve reči je upotrebljen drugi algoritam koji je treniran na znatno manjem tekstualnom korpusu. Takođe, treba napomenuti da su rezultati evaluirani poređenjem sa ocenama dobijenim od dvojice ljudi - sudija koji su određivali semantičku sličnost među rečenicama. U nekim slučajevima, bilo je neslaganja, pa je treći sudija davao krajnji sud. Ovo je interesantno imajući u vidu da je cilj ovog i sličnih istraživanja, implementacija i modifikacija algoritama koji pokušavaju da mere semantičku sličnost na isti način kao čovek, a pri tom nije do kraja jasno kao ovo funkcioniše. Dakle, glavni izazov je kako odrediti najbolju meru, dok precizna definicija te mere i dalje ostaje nepoznata.

8. Zahvalnost

Ovaj rad je delimično finansiran od strane Ministarstva nauke i prosvete Republike Srbije (projekat III44009).

LITERATURA

[1] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, Jul. 2008, pp. 1-25.

[2] M. Sahlgren, "An introduction to random indexing," *In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, 2005.

[3] S. Dumais, "Latent semantic analysis," *Annual Review of Information Science and Technology*, vol. 38, 2004, p. 188–230.

[4] K. Lund, "Hyperspace analog to language (hal): A general model of semantic representation," *Language and Cognitive Processes*, 1996.

[5] D. Rohde, "An improved method for deriving word meaning from lexical co-occurrence," *Cognitive Science*, 2004.

[6] D. Jurgens and K. Stevens, "The S-Space package: an open source package for word space models," *Association of Computational Linguistics*, 2010, pp. 30-34.

[7] W. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," *20th International Conference on Computational Linguistics*, 2004.

[8] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," *European Chapter of the Association for Computational Linguistics*, 2009, pp. 567-575.

Abstract – This paper describes software system that estimates semantic similarity of short text. Furthermore, the basic principles of the system development are explained, including algorithms for measuring lexical and semantic similarity, and an algorithm that combines these two approaches in order to obtain the best results. Phases of application development are presented as well as evaluation results. Also, at the end of the paper special emphasis is put on the further work and possible use of existing methods for the Serbian language.

SOFTWARE SYSTEM FOR MEASURING THE SEMANTIC SIMILARITY OF SHORT TEXTS

Davor Jovanović, Bojan Furlan, Boško Nikolić